

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA: CÔNG NGHỆ THÔNG TIN**

---



**BÁO CÁO THỰC HÀNH**  
**BÀI THỰC HÀNH 1**  
**Cơ sở An Toàn Thông Tin**

<b>Giảng viên:</b>	<b>Đinh Trường Duy</b>
<b>Nhóm môn học:</b>	<b>01</b>
<b>Tổ thực hành:</b>	<b>01</b>
<b>Tên sinh viên:</b>	<b>Đinh Thị Thanh Tâm</b>
<b>Mã sinh viên</b>	<b>B22DCAT253</b>

**HÀ NỘI, THÁNG 10 NĂM 2024**

# MỤC LỤC

MỤC LỤC .....	2
BÀI THỰC HÀNH: Các giao thức mạng cơ bản ( <i>labtainer network-basics</i> ) .....	4
1. Khái niệm cơ bản: .....	4
2. Các bước thực hiện:.....	4
2.1 Khám phá .....	5
2.2 ARP .....	5
2.3 TCP .....	7
3. Kết thúc bài lab và kiểm tra: .....	7
BÀI THỰC HÀNH: Phát hiện các lỗ hổng bảo mật sử dụng công cụ rà quét nmap ( <i>labtainer nmap-discovery</i> ) .....	9
1. Khái niệm cơ bản: .....	9
2. Các bước thực hiện:.....	9
3. Kết thúc bài lab và kiểm tra: .....	11
BÀI THỰC HÀNH: Sử dụng công cụ truy cập từ xa telnet ( <i>labtainer telnetlab</i> ) ..	12
1. Khái niệm cơ bản truy cập từ xa qua Telnet và SSH: .....	12
2. Các bước thực hiện:.....	12
2.1 Xác định IP của các máy.....	12
2.2 Thực hiện telnet từ máy khách vào máy chủ và đọc dữ liệu trên máy chủ .	13
2.3 Để có thể xem mật khẩu người dùng nhập khi dùng telnet cần thực hiện các bước sau: .....	13
2.4 Thực hiện ssh từ máy khách vào máy chủ và đọc dữ liệu trên máy chủ. ....	14
3. Kết thúc bài lab và kiểm tra: .....	15
BÀI THỰC HÀNH: Giới thiệu về Wireshark ( <i>labtainer wireshark-intro</i> ) .....	16
1. Khái niệm cơ bản.....	16
2. Các bước thực hiện:.....	16
2.1 Khám phá .....	16
2.2 Chạy Wireshark để thực hiện phân tích PCAP.....	16
2.3 Tìm một gói tin cụ thể.....	18
2.4 Khám phá thêm .....	20

3. Kết thúc bài lab và kiểm tra: .....	20
KẾT LUẬN.....	21

## **BÀI THỰC HÀNH: Các giao thức mạng cơ bản (*labtainer network-basics*)**

### **1. Khái niệm cơ bản:**

**Mạng máy tính trong môi trường Linux:** hoạt động dựa trên các giao thức và công cụ phổ biến để quản lý và giao tiếp giữa các thiết bị trong mạng. Các giao thức như TCP/IP, ARP, và các lệnh như ping, ifconfig, netstat, cùng các tiện ích như tcpdump là những công cụ thiết yếu để quản lý mạng trên Linux.

**Giao thức ARP (Address Resolution Protocol):** là một giao thức mạng dùng để ánh xạ địa chỉ IP (Internet Protocol) sang địa chỉ MAC (Media Access Control). Khi một máy tính cần giao tiếp với một thiết bị khác trong cùng mạng cục bộ (LAN), nó sử dụng ARP để tìm địa chỉ MAC tương ứng với địa chỉ IP mà nó muốn giao tiếp.

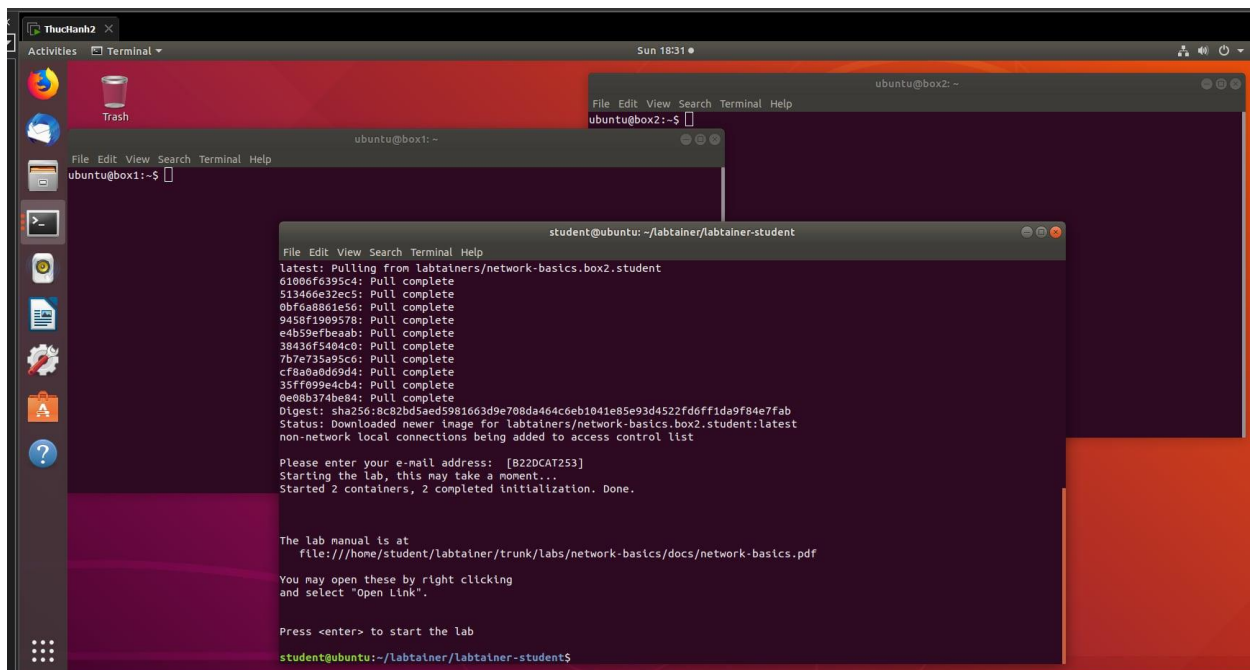
**Lệnh ping:** là một công cụ mạng cơ bản được sử dụng để kiểm tra tính kết nối giữa hai thiết bị trong mạng. Nó gửi các gói ICMP (Internet Control Message Protocol) Echo Request đến một địa chỉ IP mục tiêu và chờ phản hồi (Echo Reply).

**Cú pháp:** ping <địa\_chỉ\_IP>

- **TCP/IP (Transmission Control Protocol/Internet Protocol):** là tập hợp các giao thức được sử dụng để truyền tải dữ liệu qua mạng Internet
- **TCP:** Đảm bảo dữ liệu được truyền đi chính xác và đầy đủ, sử dụng kỹ thuật kiểm tra lỗi và truyền lại gói tin nếu cần.
- **IP:** Định tuyến và chuyển tiếp các gói tin từ nguồn đến đích thông qua các địa chỉ IP.

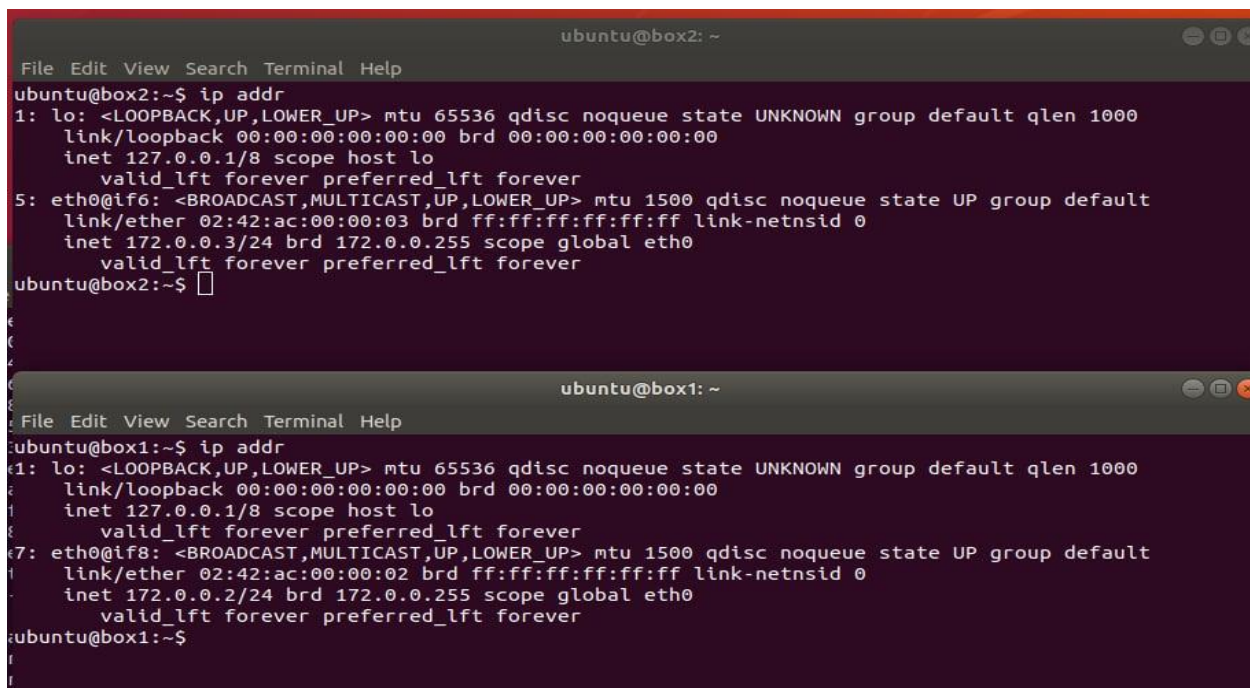
### **2. Các bước thực hiện:**

Bắt đầu khởi chạy bài thực hành, 2 terminal ảo xuất hiện: 1 terminal mang tên “box1” đóng vai trò máy tính 1 và 1 terminal mang tên “box2” đóng vai trò máy tính 2, hai máy tính kết nối với nhau qua mạng ảo có thể coi như kết nối qua dây cáp Ethernet .



## 2.1 Khám phá

Trên terminal **box1** và **box2** sử dụng lệnh: *ip addr* để kiểm tra địa chỉ IP trên cả hai máy tính :



## 2.2 ARP

Trên box2, sử dụng lệnh: *arp -a* để xem bảng ánh xạ hiện tại. Không có gì hiển thị vì bảng ARP đang trống. Khi hai máy tính của chúng ta mới khởi động, chúng không biết địa chỉ MAC của nhau.

Trên box1, khởi động chương trình tcpdump để quan sát lưu lượng mạng:

*sudo tcpdump -vv -n -e -i eth0*

```
kubuntu@box1:~$ sudo tcpdump -vv -n -e -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
01:37:36.761529 02:42:dd:5a:20:7b > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 87: (tos 0x0, ttl 255, id 57930, offset 0, flags [DF], proto UDP (17), length 73)
172.0.0.101.5353 > 224.0.0.251.5353: [bad udp cksum 0x8da7 -> 0x46d6!] 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
01:38:10.177930 82:aa:6e:a4:66:fa > 33:33:00:00:00:02, ethertype IPv6 (0x86dd), length 70: (hlim 255, next-header ICMPv6 (58) payload length: 16) fe80::80aa:6eff:fea4:66fa > ff02::2: [icmp6 sum ok] ICMP6, router solicitation, length 16
source link-address option (1), length 8 (1): 82:aa:6e:a4:66:fa
0x0000: 82aa 6ea4 66fa
01:38:44.024091 82:aa:6e:a4:66:fa > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 107: (flowlabel 0x47edc, hlim 255, next-header UDP (17) payload length: 53) fe80::80aa:6eff:fea4:66fa.5353 > ff02::fb.5353: [bad udp cksum 0x540e -> 0x806f!] 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
01:38:44.556288 02:42:dd:5a:20:7b > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 107: (flowlabel 0x771ac, hlim 255, next-header UDP (17) payload length: 53) fe80::42:ddff:fe5a:207b.5353 > ff02::fb.5353: [bad udp cksum 0xfbdc -> 0xd8a0!] 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
```

Trên box2, sử dụng lệnh ping để ping box1:

*ping 172.0.0.2 -c 2*

```
kubuntu@box2:~$ ping 172.0.0.2 -c 2
PING 172.0.0.2 (172.0.0.2) 56(84) bytes of data.
64 bytes from 172.0.0.2: icmp_seq=1 ttl=64 time=1.98 ms
64 bytes from 172.0.0.2: icmp_seq=2 ttl=64 time=0.853 ms

--- 172.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.853/1.414/1.976/0.561 ms
kubuntu@box2:~$
```

### Nhiệm vụ con 2.2.1: Tin tưởng vào phản hồi ARP

Trong hệ thống mạng, khi một thiết bị (box) gửi yêu cầu ARP để ánh xạ địa chỉ IP sang địa chỉ MAC, nó kỳ vọng rằng chỉ thiết bị đúng với địa chỉ IP đó mới phản hồi. Tuy nhiên, giao thức ARP thiếu cơ chế xác thực, do đó nó có thể dễ dàng bị lợi dụng thông qua một kỹ thuật tấn công gọi là **ARP spoofing**.

### Nhiệm vụ con 2.2.2: Giao tiếp ngoài subnet

**Subnet (mạng con)** là một phân đoạn nhỏ của một mạng lớn hơn, được chia dựa trên **mặt nạ mạng con (subnet mask)**.

Khi hai **máy** tính nằm trong **các subnet khác nhau**, chúng không thể giao tiếp trực tiếp thông qua ARP. Điều này là do ARP chỉ hoạt động trong phạm vi mạng cục bộ (LAN).



**Gateway** (thường là router) là thiết bị đảm nhiệm vai trò **chuyển tiếp gói tin** giữa các subnet khác nhau. Router hoạt động ở lớp **Layer 3 (Network Layer)** của mô hình OSI, sử dụng giao thức **IP** để định tuyến gói tin.

## 2.3 TCP

Khởi động lại tcpdump trên box1, lần này không sử dụng tùy chọn -e:

```
sudo tcpdump -vv -n -i eth0
```

```
ubuntu@box1: ~  
File Edit View Search Terminal Help  
len 6), IPv4 (len 4), Request who-has 172.0.0.3 tell 172.0.0.2, length 28  
01:42:17.394813 02:42:ac:00:00:03 > 02:42:ac:00:00:02, ethertype ARP (0x0806), length 42: Ethernet (len 6), IPv4 (len 4), Reply 172.0.0.3 is-at 02:42:ac:00:00:03, length 28  
f  
[sudo tcpdump -vv -n -i eth0  
[sudo tcpdump -vv -n -i eth0  
01:46:08.750633 02:42:dd:5a:20:7b > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 87: (tos 0x0, ttl 255, id 15511, offset 0, flags [DF], proto UDP (17), length 73)  
[ 172.0.0.101.5353 > 224.0.0.251.5353: [bad udp cksum 0x8da7 -> 0x46d6!] 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)  
01:46:22.130221 82:aa:6e:a4:66:fa > 33:33:00:00:00:02, ethertype IPv6 (0x86dd), length 70: (hlen 255, next-header ICMPv6 (58) payload length: 16) fe80::80aa:6eff:fea4:66fa > ff02::2: [icmp6 sum ok] ICMPv6, router solicitation, length 16  
[ source link-address option (1), length 8 (1): 82:aa:6e:a4:66:fa  
[ 0x0000: 82aa 6ea4 66fa  
01:47:16.027683 82:aa:6e:a4:66:fa > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 107: (flowlabel 0x47edc, hlim 255, next-header UDP (17) payload length: 53) fe80::80aa:6eff:fea4:66fa.5353 > ff02::fb.5353: [bad udp cksum 0x540e -> 0x806f!] 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)  
01:47:16.558201 02:42:dd:5a:20:7b > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 107: (flowlabel 0x771ac, hlim 255, next-header UDP (17) payload length: 53) fe80::42:ddff:fe5a:207b.5353 > ff02::fb.5353: [bad udp cksum 0xfbd4 -> 0xd8a0!] 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
```

Trên box2, khởi tạo một phiên SSH tới box1. Chúng ta sẽ không thực sự hoàn tất việc đăng nhập, chỉ đơn giản muốn xem xét phần đầu của phiên:

```
ssh 172.0.0.2
```

```
ubuntu@box2:~$ ssh 172.0.0.2  
The authenticity of host '172.0.0.2 (172.0.0.2)' can't be established.  
ECDSA key fingerprint is SHA256:ZtE8xi5Y50aUktZ/XtgjIs1c5jxYQB84Vq5ofmlgGng.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

## 3. Kết thúc bài lab và kiểm tra:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab và kiểm tra:

```
stoptlab network-basics
```

```
checkwork network-basics
```

```

status: downloaded newer image for lab00ther5/lab00ther5-grader:latest
Labname network-basics

Student          |          ssh-dump |          arp |
=====
B22DCAT253      |          Y        |          Y   |
What is automatically assessed for this lab:
  ssh-dump: Did ssh while tcpdump ran
  arp: observed ARP table with an entry for box1

```



# BÀI THỰC HÀNH: Phát hiện các lỗ hổng bảo mật sử dụng công cụ rà quét nmap (*labtainer nmap-discovery*)

## 1. Khái niệm cơ bản:

**Lỗ hổng bảo mật** (security vulnerability) là các điểm yếu, lỗi hoặc thiếu sót trong một hệ thống hoặc phần mềm mà kẻ tấn công có thể lợi dụng để xâm nhập hoặc gây thiệt hại.

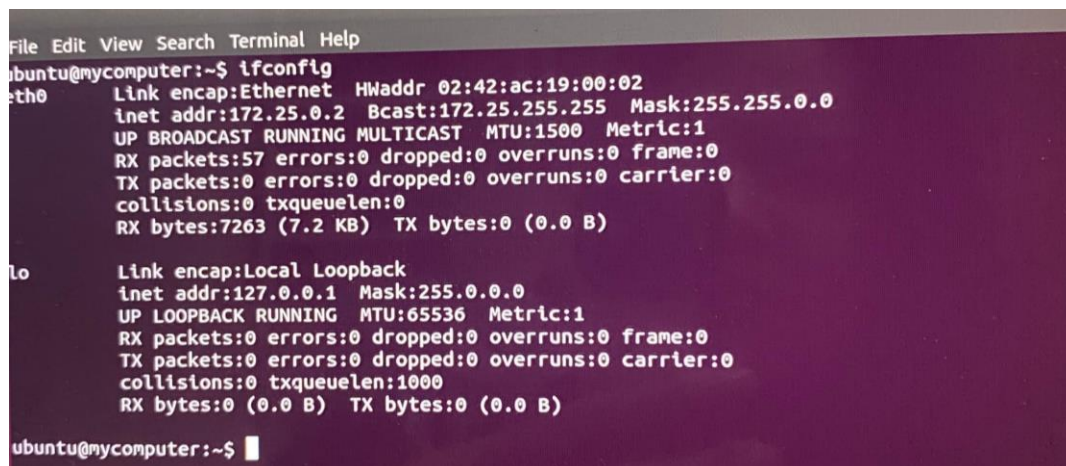
**Nmap** (Network Mapper) là một công cụ mạnh mẽ để quét và phát hiện các lỗ hổng bảo mật trong hệ thống mạng. Nmap có khả năng rà soát các cổng mở, xác định dịch vụ đang chạy trên các cổng đó, và kiểm tra các lỗ hổng bảo mật tiềm ẩn.

## 2. Các bước thực hiện:

Sau khi khởi động xong hai terminal ảo sẽ xuất hiện, một cái là đại diện cho máy khách: *mycomputer*, một cái là đại diện cho máy chủ: *friedshrimp*. Biết rằng 2 máy nằm cùng mạng LAN.

Trên terminal *mycomputer*, xác định địa chỉ IP và địa chỉ mạng LAN sử dụng lệnh :

*Ifconfig*



```
File Edit View Search Terminal Help
ubuntu@mycomputer:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:19:00:02
          inet addr:172.25.0.2  Bcast:172.25.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7263 (7.2 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ubuntu@mycomputer:~$
```

Sử dụng nmap để tìm ra địa chỉ IP của máy *friedshrimp* vì chúng cùng nằm trong mạng LAN

*nmap -sP 172.25.0.0/24*

```

ubuntu@mycomputer:~$ nmap -sP 172.25.0.0/24

Starting Nmap 7.01 ( https://nmap.org ) at 2024-09-08 15:27 UTC
Nmap scan report for mycomputer (172.25.0.2)
Host is up (0.0024s latency).
Nmap scan report for nmap-discovery.friedshrimp.student.intranet (172.25.0.5)
Host is up (0.0027s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.01 seconds
ubuntu@mycomputer:~$
ubuntu@mycomputer:~$

```

Sử dụng nmap để tìm cổng dịch vụ đang mở trên máy *friedshrimp*

*nmap -p 2000-3000 172.25.0.5*

```

ubuntu@mycomputer:~$
ubuntu@mycomputer:~$ nmap -p 2000-3000 172.25.0.5

Starting Nmap 7.01 ( https://nmap.org ) at 2024-09-08 15:47 UTC
Nmap scan report for nmap-discovery.friedshrimp.student.intranet (172.25.0.5)
Host is up (0.0033s latency).
Not shown: 1000 closed ports
PORT      STATE SERVICE
2394/tcp  open  ms-olap2

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
ubuntu@mycomputer:~$

```

Sử dụng ssh để truy cập vào máy chủ

*ssh -p 2394 172.25.0.5*

```

2394/tcp open  ms-olap2

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
ubuntu@mycomputer:~$ ssg -p 2394 172.25.0.5
-su: ssg: command not found
ubuntu@mycomputer:~$ ssh -p 2394 172.25.0.5
The authenticity of host '[172.25.0.5]:2394 ([172.25.0.5]:2394)' can't be established.
ECDSA key fingerprint is SHA256:nFDnpYXdIsAGpF1Zx0Bv8Xc83CDp5qYU2frYQvB7Pt8.
Are you sure you want to continue connecting (yes/no)?

```

Sau khi truy cập được vào máy chủ **friedshrimp** đi tìm file **friedshrimp.txt**. Mở và đọc file.

*cat friedshrimp.txt*

Đóng kết nối từ máy **mycomputer** đến **friedshrimp**. Sử dụng lệnh: “close”

```

ubuntu@friedshrimp:~$ cat friedshrimp.txt
My summary notes from the fried shrimp project:

Fried Shrimp Project: We concluded it is better to
buy than to build.

=====

Congratulations! You managed to find the summary file
for "fried shrimp" and impress Randall.
ubuntu@friedshrimp:~$ exit
logout
Connection to 172.25.0.5 closed.
ubuntu@mycomputer:~$

```

### 3. Kết thúc bài lab và kiểm tra:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab và kiểm tra:

*stoplab nmap-discovery*

*checkwork nmap-discovery*

```

student@ubuntu:~/labtainer/labtainer-student$ checkwork nmap-discovery
nmap-discovery lab is not running, looking for previous results...
Labname nmap-discovery

Student          |      nmap_count |      did_ssh |
=====|=====|=====|
B22DCAT253      |          2      |          Y   |
What is automatically assessed for this lab:
    did_ssh: SSH'd to the proper port and viewed the target file
    nmap_count: count of use of nmap

```



# BÀI THỰC HÀNH: Sử dụng công cụ truy cập từ xa telnet (*labtainer telnetlab* )

## 1. Khái niệm cơ bản truy cập từ xa qua Telnet và SSH:

**Telnet** và **SSH** là hai giao thức mạng cho phép bạn truy cập từ xa vào một máy chủ từ một máy khách để thực hiện các lệnh và thao tác trên hệ thống từ xa. Tuy nhiên, **SSH** (Secure Shell) là giao thức được ưu tiên hơn do bảo mật tốt hơn so với Telnet.

**Telnet** là một giao thức cũ để kết nối với máy chủ từ xa thông qua giao diện dòng lệnh. Tuy nhiên, Telnet không mã hóa dữ liệu, bao gồm cả mật khẩu, dẫn đến nguy cơ bị nghe lén. Do đó, nó ít được sử dụng ngày nay.

**SSH** (Secure Shell) là giao thức mã hóa để thực hiện kết nối từ xa an toàn và phổ biến nhất hiện nay.

Sự khác biệt giữa **Telnet** và **SSH**:

- **Telnet**: Không mã hóa dữ liệu, dễ bị tấn công bởi các cuộc tấn công nghe lén.
- **SSH**: Mã hóa toàn bộ dữ liệu, bao gồm cả thông tin đăng nhập, an toàn và bảo mật hơn.

## 2. Các bước thực hiện:

Sau khi khởi động xong hai terminal ảo sẽ xuất hiện, một cái là đại diện cho máy khách: *client*, một cái là đại diện cho máy chủ: *server*.

### 2.1 Xác định IP của các máy

Trên cả 2 terminal *client* và *server* sử dụng lệnh “*ifconfig*”, địa chỉ IP sẽ nằm sau “*inet addrr*”.



```
ubuntu@client:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:14:00:02
          inet addr:172.20.0.2  Bcast:172.20.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7089 (7.0 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ubuntu@client:~$

ubuntu@server:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:14:00:03
          inet addr:172.20.0.3  Bcast:172.20.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7355 (7.3 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ubuntu@server:~$
```

## 2.2 Thực hiện telnet từ máy khách vào máy chủ và đọc dữ liệu trên máy chủ

Trên máy khách sử dụng telnet để kết nối với máy chủ thông qua địa chỉ IP:

*telnet 172.20.0.3*

```
ubuntu@client:~$ telnet 172.20.0.3
Trying 172.20.0.3...
Connected to 172.20.0.3.
Escape character is '^['.
Ubuntu 16.04.4 LTS
server login: █
```

Để kết nối, nhập “ubuntu” cho cả username và password (chú ý: trong khi nhập mật khẩu sẽ không có ký tự nào được hiển thị).

```
Connection closed by foreign host.
ubuntu@client:~$ telnet 172.20.0.3
Trying 172.20.0.3...
Connected to 172.20.0.3.
Escape character is '^['.
Ubuntu 16.04.4 LTS
server login: ubuntu
Password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@server:~$
```

Sau khi đăng nhập vào máy chủ, thực hiện việc đọc tệp có sẵn chứa đoạn mật mã của sinh viên:

*cat filetoview.txt*

Sau đó thoát khỏi phiên telnet trên máy khách thông qua lệnh “exit”:

```
ubuntu@server:~$ cat filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (telnet-server) container
#
# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 1a939eb054d2323796f3d7b11b0b1db2
ubuntu@server:~$ exit
logout
Connection closed by foreign host.
ubuntu@client:~$
```

## 2.3 Để có thể xem mật khẩu người dùng nhập khi dùng telnet cần thực hiện các bước sau:

Trên server, hãy dùng câu lệnh sau để chạy tcpdump, giúp hiển thị các gói tin TCP:

*sudo tcpdump -i eth0 -X tcp*

```
ubuntu@server:~$ sudo tcpdump -i eth0 -X tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Thực hiện phiên telnet trên máy khách, khi được nhắc nhập mật khẩu, nhập "abc123" (như đã biết mật khẩu này không chính xác). Khi nhập từng chữ cái của mật khẩu, quan sát tcpdump của lưu lượng truy cập. Lưu ý rằng mỗi gói tin khác là một "ack", sinh viên có thấy mật khẩu không? Sinh viên nhận thấy điều gì?

```
ubuntu@client:~$ telnet 172.20.0.3
Trying 172.20.0.3...
Connected to 172.20.0.3.
Escape character is '^J'.
ubuntu 16.04.4 LTS
server login: ubuntu
Password:
Login incorrect
server login:
Login timed out after 60 seconds.
Connection closed by foreign host.
```

```
1 9aed 97e4 7b84 59b7 9ae4 6135 1982 P....{Y...a5..
2 ef96 62c1 3e9d 3ceb c76a 8ac4 79bd ...b>.<....y.
3 6d6a b438 0000 0020 43da 75ff 36e2 x.mj.8....C.u.6.
4 852c 3bc6 f551 dbf7 6061 52d4 3c1b %...;...Q...ar.<.
5 b59c d29c 2f5a d177 0000 0064 0000 @....Z.w...d..
6 6563 6473 612d 7368 6132 2d6e 6973 ..ecdsa-sha2-ntS
7 3235 3600 0000 4900 0000 2100 633e tp256...I...!>
8 82cd 40c5 d56a fb3c f42c 9ecf f632 ^..@.j<...2
9 a84c 07f0 8994 cdb4 5e6b e6e0 0000 ..L.....K....
10 5351 969a 9adf deda dadd 48bd 6bda ..SQ.....H.k.
11 5c20 2784 1d2d bfaa 29df b968 2fa5 .I\'......).h/.
12 0000 0000 0000 0000 0000 0000 000c .....
13 0000 0000 0000 0000 0000 0000 bed9 f789 .....
14 f1c0 8c90 3e85 9d25 0de5 04c4 6d92 .....>..M.
15 2d87 3e20 30ba 6301 d801 9cbd 8758 ...>0.C.....X
16 6a83 cfcc bb57 0fcd ecac 4d19 347d pSj...Wo...M.4}
17 9c14 9874 f9cd 2130 bc03 af4b 31dd d....t..l0...K1.
18 7604 ae8a bb67 01b2 79fc eb15 4c14 (.v...g.y...L.
19 lnnetlab.client.student.some_network.59062 > server.ssh: Flags [..], ack 1382, wt
20 op,TS val 1074437067 ecr 3331434903, length 0
21 0034 b459 4000 4000 2e3d ac14 0002 E..4.Y0.0.1=...
```

## Trả lời:

Telnet là một giao thức không mã hóa, nghĩa là tất cả dữ liệu (bao gồm tên đăng nhập và mật khẩu) đều được truyền qua mạng dưới dạng văn bản thuần túy (plaintext). Khi quan sát tcpdump, sẽ thấy từng gói tin chứa các ký tự của mật khẩu "abc123" được gửi đến máy chủ. Mỗi ký tự của mật khẩu sẽ được truyền riêng biệt và sẽ xuất hiện dưới dạng gói tin khác nhau trên luồng mạng.

## Nhận xét:

- **Có thể thấy mật khẩu:** Vì Telnet không mã hóa dữ liệu, có thể dễ dàng thấy mật khẩu "abc123" dưới dạng văn bản thuần túy khi phân tích gói tin.
- **Nhận thấy rằng:** khi sử dụng Telnet, dữ liệu quan trọng như mật khẩu có thể dễ dàng bị đánh cắp nếu không có biện pháp bảo mật như sử dụng các giao thức an toàn hơn như SSH.

## 2.4 Thực hiện ssh từ máy khách vào máy chủ và đọc dữ liệu trên máy chủ.

Trên máy khách sử dụng ssh để kết nối với máy chủ thông qua địa chỉ IP:

*Ssh 172.20.0.3*

```
ubuntu@client:~$ ssh 172.20.0.3
The authenticity of host '172.20.0.3 (172.20.0.3)' can't be established.
ECDSA key fingerprint is SHA256:nFDnpYXdisAGpF1Zx0BV8Xc83CDp5qYU2frYQvB7Pt8.
Are you sure you want to continue connecting (yes/no)?
```

Sau khi đăng nhập vào máy chủ, thực hiện việc đọc tệp có sẵn chứa đoạn mật mã:

*cat filetoview.txt*



Sau đó thoát khỏi phiên ssh trên máy khách thông qua lệnh “exit”.

### 3. Kết thúc bài lab và kiểm tra:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab và kiểm tra:

*stoplab telnetlab*

*checkwork telnetlab*

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork telnetlab
telnetlab lab is not running, looking for previous results...
Labname telnetlab

Student          | telnetview | sshview | failed_login |
=====|=====|=====|=====|
B22DCAT253      |           Y |         Y |             Y |
What is automatically assessed for this lab:
    failed_login: Failed login as expected.
    telnetview: viewed file from telnet
    sshview: viewed file from ssh
```

# BÀI THỰC HÀNH: Giới thiệu về Wireshark (*labtainer wireshark-intro*)

## 1. Khái niệm cơ bản

### 2. Các bước thực hiện:

Sau khi khởi động bài thực hành: *labtainer wireshark-intro*

### 2.1 Khám phá

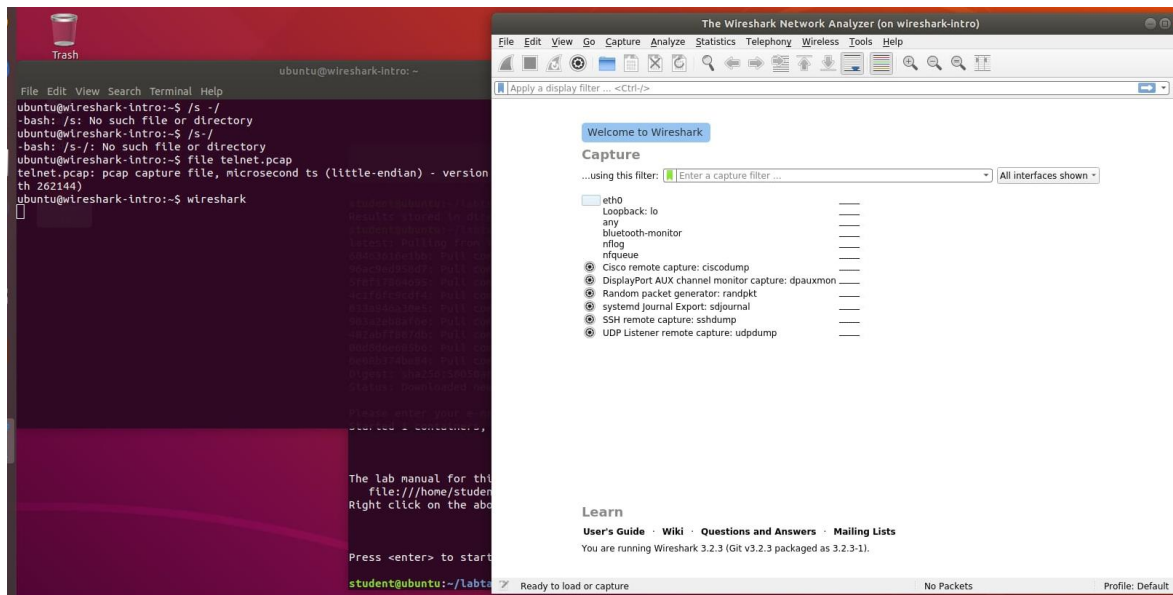
Sử dụng lệnh `ls -l` để xem nội dung của thư mục trong terminal đã mở. Tập `telnet.pcap` chứa lưu lượng mạng sẽ thực hiện để phân tích. Để xem thông tin về tệp sử dụng lệnh:

*file telnet.pcap*

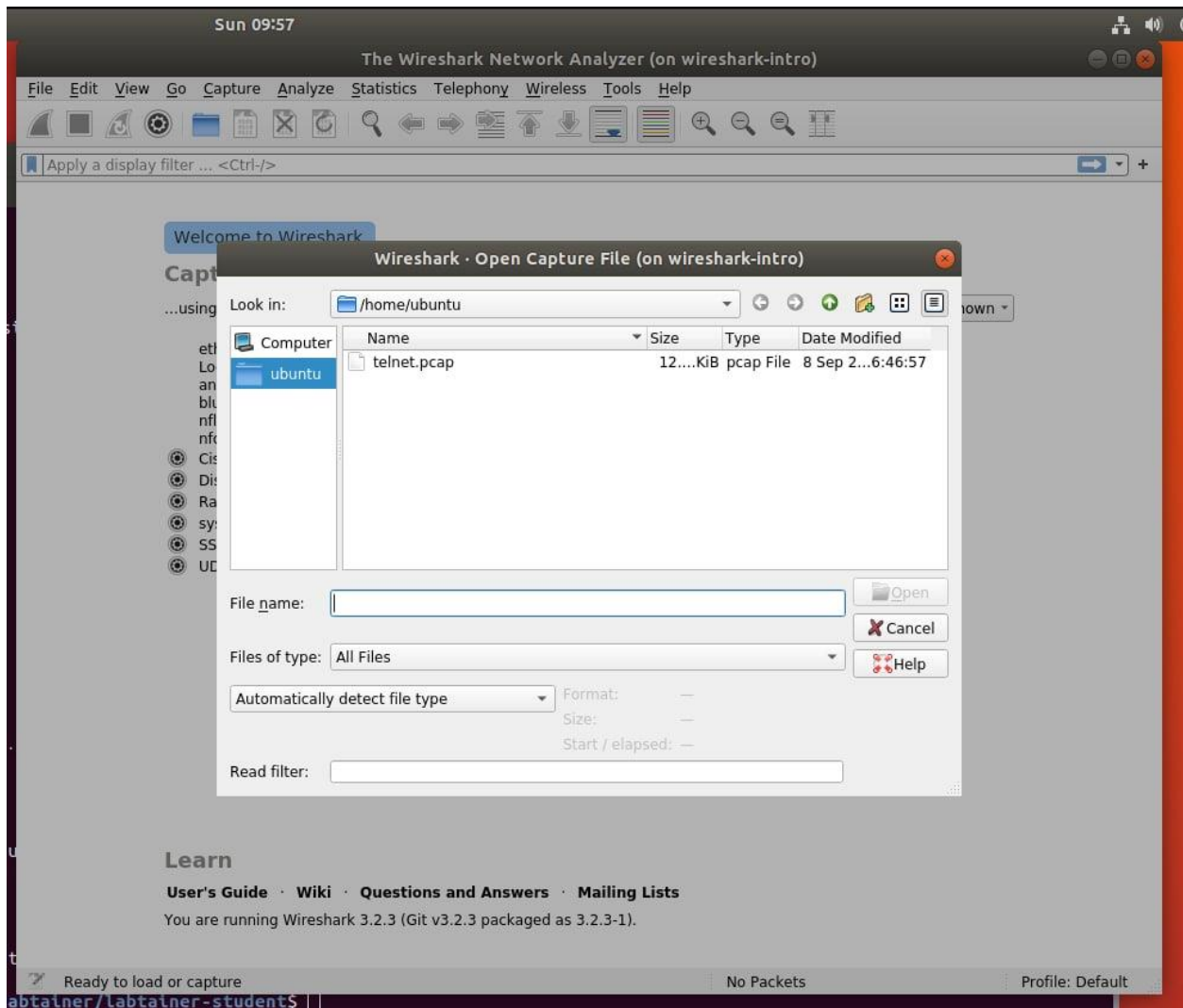
```
ubuntu@wireshark-intro:~$ file telnet.pcap
telnet.pcap: pcap capture file, microsecond ts (little-endian) - version 2.4 (Ethernet, capture length 262144)
ubuntu@wireshark-intro:~$
```

### 2.2 Chạy Wireshark để thực hiện phân tích PCAP

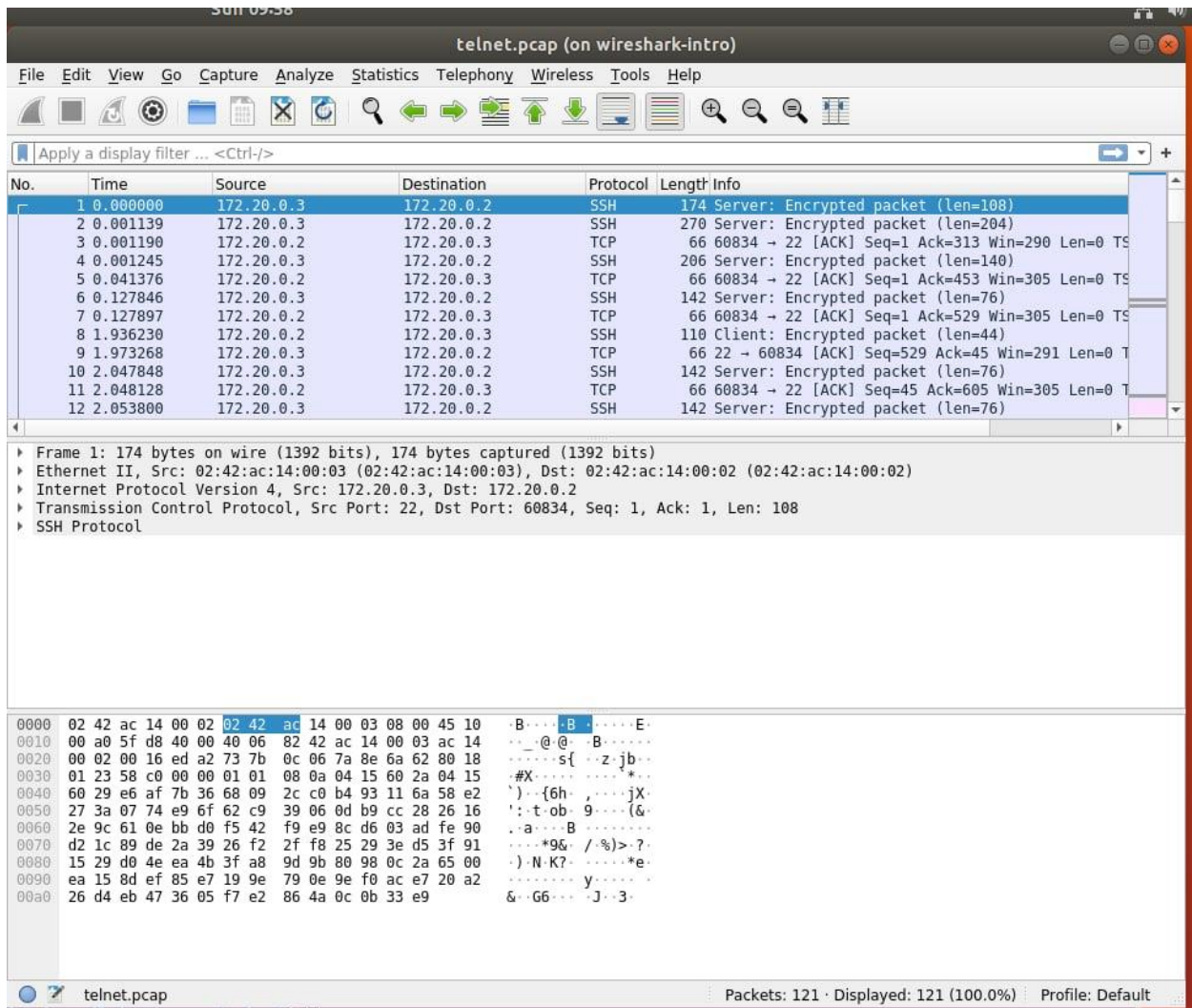
Khởi động Wireshark bằng cách sử dụng lệnh `wireshark`.



Sau đó sử dụng “File->Open” để mở tệp *telnet.pcap*



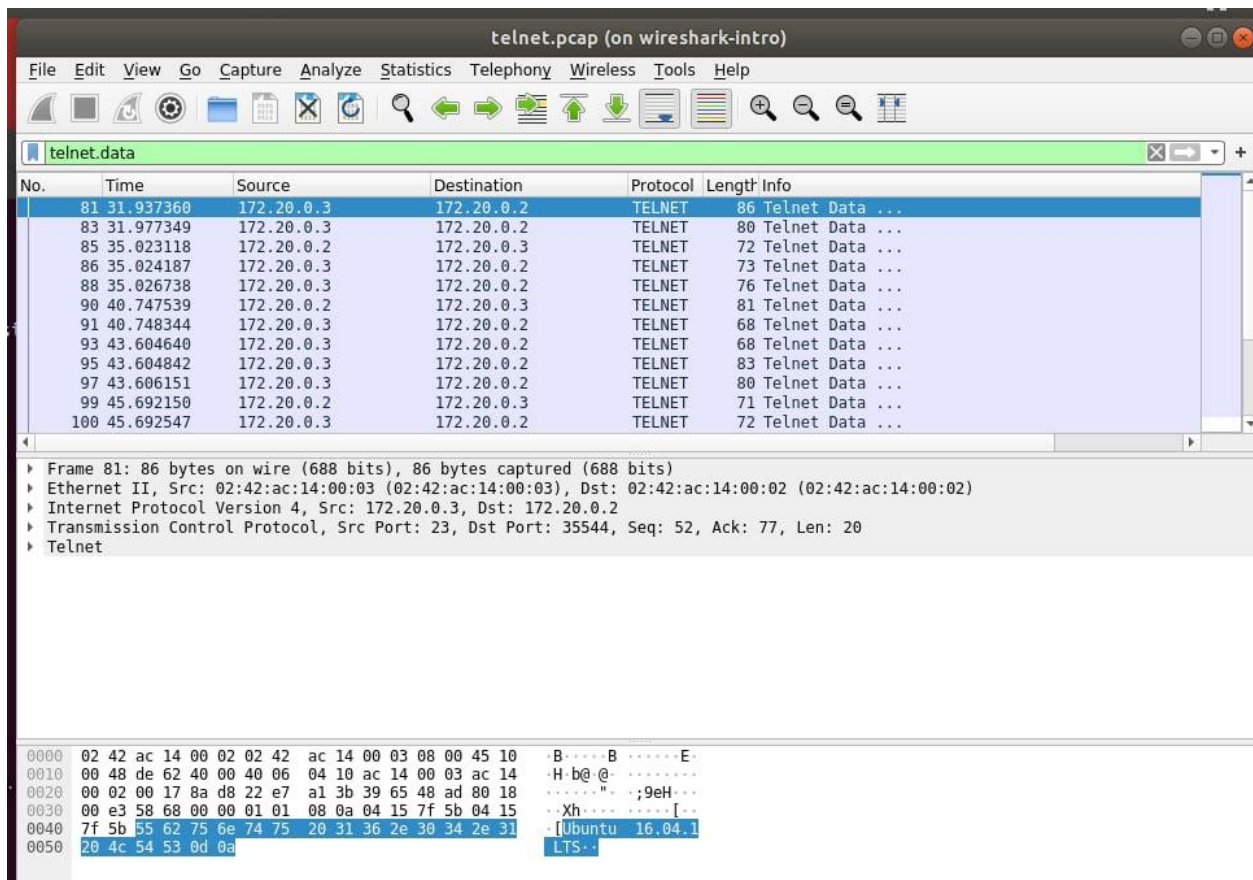
Sau khi tệp *telnet.pcap* được mở ra



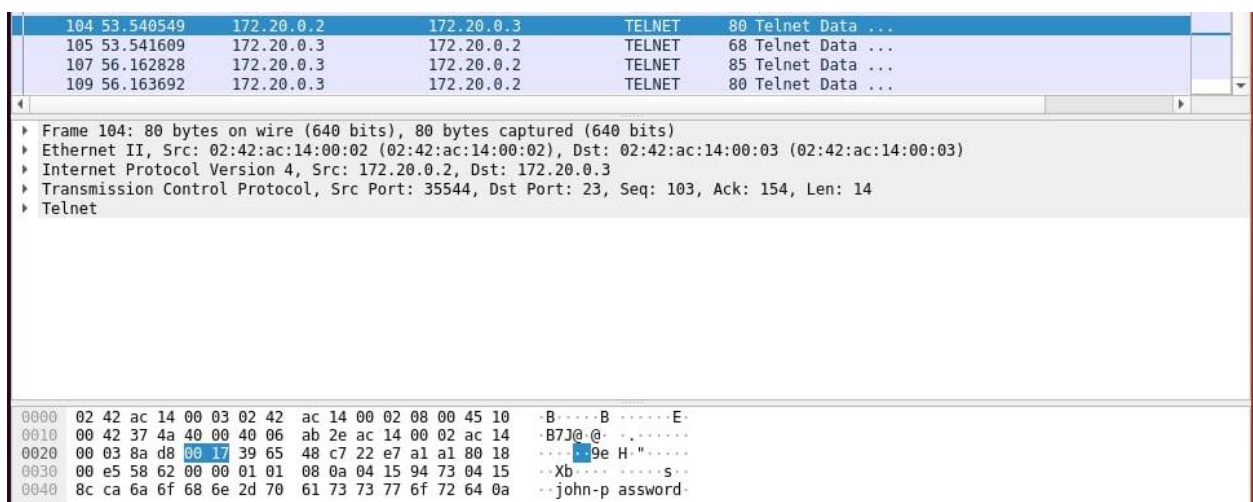
## 2.3 Tìm một gói tin cụ thể

Xác định gói tin duy nhất chứa mật khẩu được cung cấp khi người dùng cố gắng sử dụng Telnet để đăng nhập với tư cách người dùng "john".

Gợi ý: Nếu nhập telnet.data vào trường "Add a display filter", công cụ sẽ chỉ hiển thị các gói dữ liệu Telnet. Nhấn Enter để áp dụng bộ lọc.

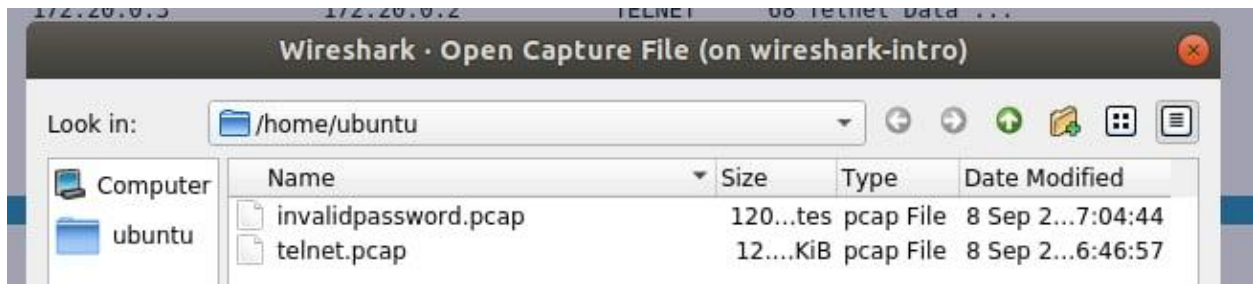


Khi tìm thấy gói tin duy nhất chứa mật khẩu không hợp lệ, sử dụng File=>Export specified packets để lưu gói tin duy nhất đã tìm thấy. Lưu gói tin duy nhất này dưới dạng *invalidpassword.pcap*. Hãy chắc chắn chọn nút radio "Selected packets only" trong hộp thoại Export và chắc chắn đặt tên tệp chính xác.



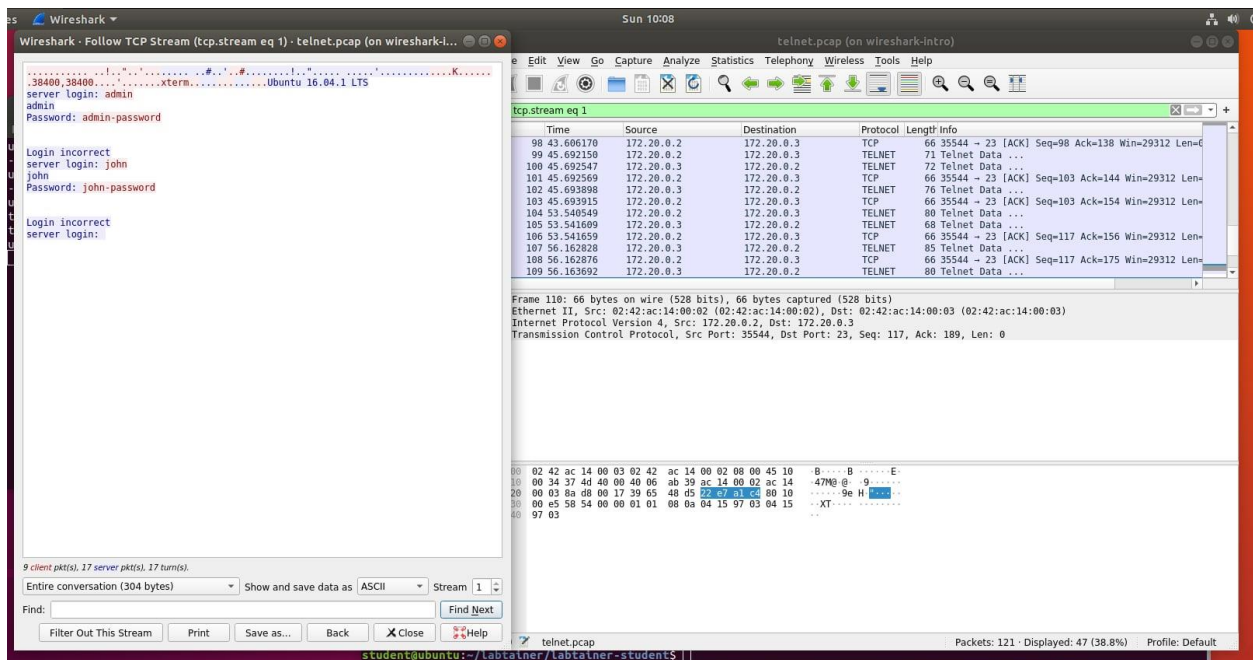
Sau khi sinh viên lưu gói tin, sinh viên có thể sử dụng File=>Open để mở tệp pcap mới của mình để xác nhận nó chứa gói tin chính xác.





## 2.4 Khám phá thêm

Xem qua các gói tin khác và thử nghiệm với các bộ lọc. Thử chọn một trong các gói TELNET và sử dụng chức năng Analyze=>Follow=>TCP stream để xem toàn bộ cuộc trò chuyện TELNET.

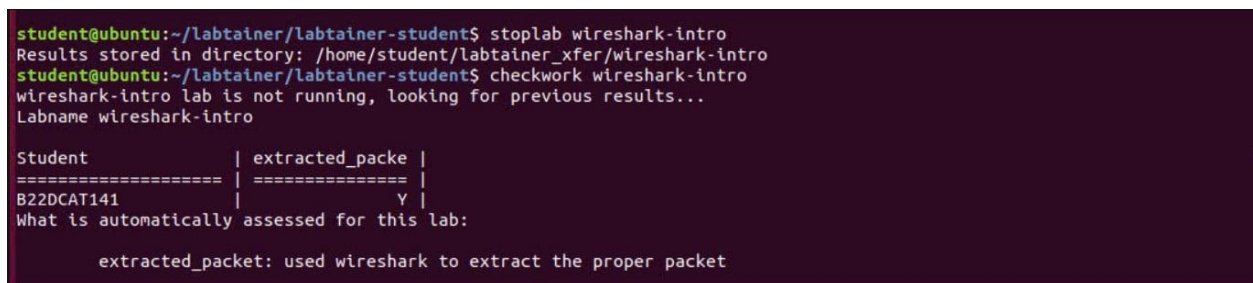


## 3. Kết thúc bài lab và kiểm tra:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab và kiểm tra:

*stoplab wireshark-intro*

*checkwork wireshark-intro*





## KẾT LUẬN

1. **Hiểu biết về các giao thức mạng cơ bản:** Qua các phần thực hành, được tiếp cận và thao tác với các giao thức mạng cơ bản như TCP/IP, ARP, cũng như các lệnh kiểm tra mạng phổ biến như ping, ifconfig, và netstat. Điều này giúp nắm rõ cách thức kết nối và trao đổi dữ liệu trong mạng.
2. **Phát hiện lỗ hổng bảo mật với Nmap:** sử dụng công cụ Nmap để phát hiện các cổng mở và lỗ hổng bảo mật trong hệ thống, giúp hiểu rõ cách mà hacker có thể khai thác lỗ hổng từ các dịch vụ không an toàn, đồng thời nâng cao ý thức về bảo mật mạng.
3. **Thực hành với Telnet và SSH:** Phần thực hành với Telnet giúp nhận thức rõ sự khác biệt giữa giao thức không mã hóa và có mã hóa. Bài học nhấn mạnh rằng việc sử dụng Telnet trong môi trường mạng không an toàn có thể gây rủi ro bảo mật nghiêm trọng, và SSH là lựa chọn an toàn hơn cho truy cập từ xa.
4. **Phân tích gói tin với Wireshark:** Qua công cụ Wireshark, có thể quan sát các gói tin mạng, từ đó thấy được cách dữ liệu truyền tải trong mạng có thể bị thu thập và phân tích. Đây là một bài học quan trọng về bảo mật thông tin trong quá trình truyền dữ liệu.
5. **Ý nghĩa thực tiễn của bảo mật mạng:** Thông qua bài thực hành này, đã nhận thức rõ hơn về tầm quan trọng của bảo mật trong việc triển khai và quản lý hệ thống mạng hiện đại, hiểu rõ về những công cụ cần thiết để bảo vệ và duy trì an toàn cho các hệ thống.